

Microsoft[®] C 5.0

优 化 编 译

用户指南

(第一册)

中国科学院 科海培训中心
软 件 所

Microsoft C 5.0

优化编译用户指南

(第一册)

译 校 者 序 言

翻 译：宗丽苹 吴 倩 郭继明 鲁 倩 王 越 徐砥祥 陈 林 彭怀宇
王淑平 何 骏 郭瑞阳 萧燕林

Microsoft C 5.0优化编译语言系统是美国87年国家标准，具有效率高、功能强的特点。为了迎合广大热心者的要求，配合国内用户高水平开发工作，由中国科学院386微机研究、开发组直接组织编译了此书共九册。虽然译校者大多是软件专业的研究生及工作多年的高资历工作人员，但由于印排仓促，审校时间紧迫，难免会有这样或那样的错误。望读者能给以谅解与指正。

参与本书印排工作的除中国科学院软件所、计算所、自动化所之外，还有祥云电脑公司、海声软件开发公司，故上述单位共有此中文资料版权。任何其它单位不得随意翻印此书。

目 录

第一章 引言	(1)
1.1 概述	(1)
1.2 关于本手册的使用	(1)
1.3 新的特征	(2)
1.4 字体和符号的约定	(4)
1.5 进一步了解C的书	(5)
第二章 初启	(7)
2.1 引言	(7)
2.2 后备磁盘	(7)
2.3 磁盘内容	(7)
2.4 SETUP 程序	(8)
2.5 了解编译软件	(16)
2.6 使用 RAM 磁盘	(22)
2.7 转换现存C程序	(22)
2.8 编译程序命令行示范	(22)
2.9 实际操作阶段	(24)
第三章 用 CL 命令进行编译	(26)
3.1 引言	(26)
3.2 基础: 编译和链接C程序	(26)
3.3 使用 CL 选择项	(29)
3.4 控制二进制方式和正文方式	(70)
第四章 用 CL 命令进行链接	(71)
4.1 引言	(71)
4.2 链接过程中的缺省方式	(71)
4.3 传递链接程序信息: /Link 选择项	(71)
4.4 链接程序选择项	(74)
第五章 在MS-DOS上运行C程序	(78)
5.1 引言	(78)
5.2 给程序传递命令行数据	(78)

5.3 返回一个退出码	(81)
第六章 存储模型的使用	(83)
6.1 引言	(83)
6.2 Near, Far 和 Huge 寻址	(83)
6.3 使用标准内存模型	(84)
6.4 使用关键字 near, far 和 huge	(87)
6.5 建立特定的内存模型	(93)
6.6 设置数据阙	(96)
6.7 命名模块和段	(96)
6.8 说明正文和数据段	(97)
第七章 控制浮点数学运算	(98)
7.1 引言	(98)
7.2 数学程序包概述	(98)
7.3 浮点选择项(/FP)的选择	(99)
7.4 浮点选择项的库考虑	(102)
7.5 浮点选择项之间的相容性	(105)
7.6 使用 No 87 环境变量	(105)
7.7 用户计算机不与 IBM 兼容怎么办	(106)
第八章 提高程序速度	(107)
8.1 引言	(107)
8.2 使用寄存器变量	(107)
8.3 优化选择项和优化杂注	(108)
8.4 选择函数调用约定	(110)
8.5 大型数据模型的效率	(110)
8.6 大型代码模型的效率	(111)
附录	(113)
A 退出码的使用	(113)
A.1 引言	(113)
A.2 MS-DOS 批文件的退出码	(113)
A.3 编辑程序的退出码	(113)
B 从以前编译程序版本转换	(114)
B.1 引言	(114)
B.2 5.0版本与4.0版本的不同	(114)
B.3 4.0版本与3.0版本的不同	(119)
C 编写可移植程序	(122)

C.1	引言.....	(122)
C.2	程序的可移植性.....	(123)
C.3	机器硬件.....	(123)
C.4	编译程序的差别.....	(127)
C.5	环境的差别.....	(130)
C.6	数据的可移植性.....	(131)
C.7	类型长度小结.....	(131)
C.8	字节顺序小结.....	(132)
D	编写 ROM 程序.....	(133)
D.1	引言.....	(133)
D.2	依赖于 MS-DOS 的库程序.....	(133)
D.3	浮点数学支持.....	(134)
D.4	修改启动程序.....	(135)
E	错误信息.....	(136)
E.1	引言.....	(136)
E.2	命令行错误信息.....	(137)
E.3	编译错误信息.....	(140)
E.4	运行时错误信息.....	(161)
	术语表.....	(164)

第一章 引言

1.1 概述

C 语言是一种强有力的通用程序设计语言，它可以产生有效、紧凑和可移植的代码。适用于 MS-DOS 操作系统的 Microsoft C 优化编译程序，是对 C 语言的作者 Brian W. Kernighan 和 Dennis M. Ritchie 在《The C Programming Language》中定义的 C 语言的一个完全实现。Microsoft 公司积极参与了 ANSI(American National Standards Institute) 对于 C 语言美国国家标准的研制。Microsoft C 的这一版本充分照顾了这一即将出现的最新 C 标准，并努力与之保持一致。

Microsoft C 提供了几个重要特性来帮助你提高 C 程序的效率。你可以在五种标准存储模型（小型、中型、紧凑型、大型和超大型）中进行选择，建立最适合于你的程序的数据存储与代码存储的组合。为了使用灵活和获得可能更高的效率，Microsoft C 优化编译程序允许你通过在程序中使用特定的说明来“混”用不同的存储模型。

C 语言本身不提供象输入输出和字符串处理这样的标准特性。这些特性由 Microsoft C 优化编译程序的运行时库函数完成。由于那些需要与操作系统打交道的函数（例如，输入和输出）逻辑上与语言本身无关，所以 C 语言特别适于产生可移植代码。

使用 MS-DOS 和 XENIX 系统的公共运行态库，能进一步提高 Microsoft C 程序的可移植性。使用库中的子程序，你可以很容易地将程序从 XENIX 开发环境移植到 MS-DOS 机器上，或进行相反的移植。关于 MS-DOS 和 XENIX 公共运行态库的详细说明，见《Microsoft C 运行态库参考手册》（包括在本套资料中）的附录 B。

与其它程序设计语言相比，C 语言的数据转换和非标准结构非常灵活。Microsoft C 优化编译程序提供了多级警告，以帮助你灵活地掌握这套软件。在程序开发的早期阶段，可以使用编译程序的全部警告能力，来发现错误和无意识的数据转换。有经验的 C 程序员可以用较低级别的警告，对需要使用非标准结构的程序进行检查。关于这个特性，见 3.3.11.2 节。

1.2 关于本手册的使用

本手册向用户解释如何在 MS-DOS 系统中使用 Microsoft C 优化编译程序去编译、链接和运行 C 程序。本手册假定用户熟悉 C 语言和 MS-DOS，并且知道如何在系统中创建和编辑 C 源文件。

注：

MS-DOS 和 PC-DOS 是基本上相同的操作系统，除它们有重大区别的地方，Microsoft 手册用 MS-DOS 这个词表示这两个系统。

如果你对C语言有疑问,请看本套资料中的《Microsoft C简明手册》。《Microsoft C运行态库参考手册》罗列了可在程序中使用的库子程序。Microsoft CodeView和实用程序手册说明了如何使用CodeView符号调试程序和Microsoft C优化编译程序软件包中提供的其它实用程序。《Microsoft混合语言程序设计指南》说明了如何把用Microsoft C、Microsoft FORTRAN、Microsoft Pascal和Microsoft BASIC编写的模块混合起来使用。关于C语言程序设计的详细情况,参见1.5节,“进一步了解C的书。”

下表给出了《Microsoft C优化编译程序用户指南》中其余各章的简要说明:

第二章“启动系统”概述了本编译软件的安装和组织。本章解释如何通过定义环境变量来为编译程序建立运行环境,并且包括了一段实际操作过程以使用户熟悉Microsoft C优化编译程序。

第三章“用CL命令进行编译”讨论了使用CL编译驱动程序来编译程序的过程。本章描述了在预处理、编译和文件输出控制中通常使用的各种选择项。

第四章“用CL命令进行链接”说明了如何使用CL命令链接目标文件。本章解释链接程序如何查找库及如何在链接中指定库,还描述了用于C程序的各种链接程序选择项。

第五章“在MS-DOS上运行C程序”,解释如何运行可执行程序文件,并讨论与C的MS-DOS实现有关的特性。本章还讲述了如何在程序执行时从MS-DOS传送数据给程序,以及如何从用户的程序返回一个退出码,传给MS-DOS。

第六章“存储模型的使用”,描述了管理存储模型的各种方法。这些方法对于编写代码或数据多于64 K的大程序是有用的。本章还讨论了“混合模型”程序设计(五种标准存储模型的组合特性)。

第七章“控制浮点数学运算”,描述了控制Microsoft C程序如何处理浮点运算的CL命令选择项,及支持浮点运算的库。

第八章“提高程序速度”给出了使程序达到最大速度的建议和提示。

附录A“使用退出码”列出了Microsoft C优化编译程序产生的退出码。本章还简短讨论了在程序维护实用程序MAKE的描述文件中和批文件中如何使用退出码。

附录B“从以前编译程序版本的转换”概括了Microsoft C 5.0版和以前版本的差异。本附录还给出了把早期版本的程序转换为5.0版可接受格式的指南。

附录C“编写可移植程序”列出了C语言一些在实现有关的特点,并提出了增加程序可移植性的建议。

附录D“为只读存储器编写程序”,给出了为将写入只读存储器的程序修改启动代码并启动浮点支持的信息。

附录E“错误信息”列出并描述了由Microsoft C优化编译程序和CL命令行产生的错误信息。以及可执行C程序产生的运行时错误信息。

1.3 新的特征

在Microsoft C优化编译程序5.0版本中增加了几个有用的新特性。本节概述了这些在4.0版本基础上增加的新特性。关于5.0版本4.0版及以前版本的差异,请见附录B“对以前编译程序版本的转换。”

新的特性列表如下:

特性 **说明**

Microsoft Quick C 编译程序 SETUP 程序 组合运行 态库	Microsoft Quick C 编译程序附随在 Microsoft C 优化编译程序的 5.0 版本中。Microsoft Quick C 提供的集中式程序设计环境包括编辑程序、编译程序、调试程序、程序维护实用程序。 自动安装 Microsoft C 优化编译程序的批处理文件。 由安装程序建立的组合运行态库, 包括标准库支持和浮点数学支持。
--	--

新的 CL 选择项

选择项

动

作

/Oi	生成库函数的内部形式
/Ol	允许循环优化
/Op	对浮点数学运算的结果强制一致性精度。
/qc	指定用 Microsoft Quick C 编译程序进行编译
/Se	指定源列表的行宽
/Sp	指定源列表的每页行数
/Ss	说明源列表的子标题
/St	说明源列表的标题
/Tc	说明不带 .C 扩展名的文件为 C 源文件
/Zp	指定按给定字节限制进行结构紧缩

Const 关键字 说明一个值在程序执行期间不会改变。

新的杂注

杂注

动作

alloc-text	对分配给定函数的代码段命名
function	禁止特定函数的内部函数生成
intrinsic	说明生成该函数的内部形式
loop-opt	对程序进行局部循环优化
pack	说明结构压缩的字节限制
same-seg	提供编译程序用于执行优化时的远数据分配信息

新的 显示链接过程的有关信息。详见 Microsoft CodeView 和实用程序手册的 /INFORMATLON 12.2.3 节。

链接程序

选择项

语言变化

对 C 语言的语法和语义做了几处修改, 以适合 ANSI “C 语言标准”的最近更新。详见附录 B “从以前编译程序版本的转换”和《Microsoft

C 语言手册》的附录 A。

新的库函数

支持所有按 ANSI 标准定义的库函数, 但不支持那些为支持国际性语言而增加的函数。修改并增强了一些已有函数。还增加了一组图形函数。详见附录 B “从以前编译程序版本转换” 和《Microsoft C 运行态库参考手册》。

1.4 字体和符号的约定

在本手册中使用以下符号约定:

约定	说明
程序字体	这是一种在屏幕上显示或由打印机打印的字体。用该字体模拟在屏幕上显示或由打印机打印的信息。如下列命令行就以该字体印刷: <code>CL /FoOUT.OBJ/DTRUE=1 FILE.C</code> 在正文中讨论到这一命令行时, 在其中出现的项如 OUT.OBJ 也以该字体印刷。
小写黑体字 (英文)	小写英文黑体字表示在源程序中严格出现的 C 语言成份。正常出现的小写英文黑体字正文包括操作符、关键字、库函数名、命令、选择项和预处理命令。举例如下: <code>+ = #if defined() int</code> <code>if /Fa fopen</code> <code>main sizeof</code>
大写英文黑体字	大写英文黑体字用于可执行文件名、本编译软件包提供的文件名、环境变量、符号常量及宏名字。在 MS-DOS 级键入的命令也用大写英文黑体字给出如 SET, 还有象 CL、LINK 和 LIB 的程序名。不过实际输入这些命令行不是必须用大写字母。
斜体字	在命令行、选择项说明和正文中的斜体字表示信息的类型, 并且该信息不可缺省。例如下面的选择项: <code>/Hnumber</code> 注意, number (数字) 是斜体字, 指出了/H选择项的一般形式。在实际命令中, 用户要在 number 出现的地方提供一个具体数字。
竖省略号	在举例程序中使用竖省略号表示程序的一部分被省略了。例如下面程序段, 两个语句之间的省略号表示其中的程序行存在但被省略了: <code>count = 0;</code> ● ● ● <code>*pc++;</code>
黑体方括号	黑体方括号用于在命令行和选择项说明中括起可选择域。例如下面的选择

项说明:

`/D identifier [= string]`

斜体字 `identifier` (标识符) 要求用户使用 `/D` 选择项时必须提供一个标识符。外层方括号表示标识符后不一定必须出现等号(=)和字符串。内层方括号表示, 在等号后不一定必须提供字符串, 但若已经提供了字符串则必须同时提供等号。

普通方括号用于C语言的数组说明和下标表达式。例如, `a[10]` 是在C下标表达式中使用普通方括号的例子。

省略号

水平省略号用于语法举例中, 表示具有相同形式的更多的项可以输入。下面语法说明例子中, 在 `PATH` 命令中可说明几个路径:

`PATH [=] path [, path] ...`

花括号和
竖杠

花括号和竖杠表示你可在2个或多个项中进行选择。花括号括住所有选择, 而竖杠把选择分开。除非所有项都用粗方括号括住, 否则你必须选择其中一项。

例如, `/W`(警告级)具有下面语法:

`/W{0|1|2|3}`

你可使用 `/W1`、`/W2` 或 `/W3` 给出不同的级别的警告信息。`/W0` 屏幕警告信息。

双引号

双引号将在正文中定义的术语划分出来。例如, 术语 `far` 出现在双引号中标记着是第一次被定义。

一些C构造也使用双引号。C中使用的双引号是 `" "` 而不是 `“ ”`。例如, 举例用的C字符串具有下面形式:

`"abc"`

键 + 键

一般大写字母表示键和键序列的名字, 例如 `ENTER` 和 `CTRL+c`。用加号隔开的大写字母键名表示该键序列必须同时按下(如 `CTRL+c`)。

1.5 进一步了解C的书

本套文档资料为程序员提供了完整的Microsoft C参考手册。但它们并不教你如何用法进行程序设计。如果你不熟悉C语言或对程序设计缺乏经验, 可阅读下列参考书中的一本或几本以熟悉C语言:

Hancock, Les, and, Morris Krieger. *The C Primer*. New York: McGrawHill Book Co., Inc., 1982.

Hansen, Augie. *Proficient C*. Bellevue, Washington: Microsoft Press, 1986.*

Harbison, Samuel P., and Greg L. Steele. *C: A Reference Manual*.

Englewood Cliffs, New Jersey: Prentice-Hall Software Series, 1987.

Kernighan, Brian W., and Dennis M. Ritchie. *The C Programming Language*.

Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1978.

Kochan, Stephen. *Programming in C*. Hasbrouck Heights, New Jersey: Hayden Book

Company, Inc., 1983.

Plum, Thomas. *Learning to Program in C*. Cardiff, New Jersey: Plum Hall, Inc., 1983.

Schildt, Herbert. *C Made Easy*. Berkeley, California: Osborne McGraw Hill, 1985.

Schustack, Steve. *Variations in C*. Bellevue, Washington: Microsoft Press, 1985.

这些书列出来仅为了方便用户。除自己的出版物, Microsoft公司并不特意推荐这些书。

1.5 进一步学C的书

本章介绍的书籍是供那些已经学完了完整的Microsoft C参考手册, 但还并不熟悉如何应用C语言进行程序设计。如果你对C语言或程序设计缺乏经验, 可阅读下列各书中的一本或几本以熟悉C语言。

Hancock, Les, and Morris Kreiger. *The C Primer*. New York: McGraw-Hill Book Co., Inc., 1983.

Hansen, Augie. *Proficient C*. Bellevue, Washington: Microsoft Press, 1985.

Hardison, Samuel F., and Greg I. Steele. *C: A Reference Manual*.

Englewood Cliffs, New Jersey: Prentice-Hall Software Series, 1987.

Kernighan, Brian W., and Dennis M. Ritchie. *The C Programming Language*.

Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1983.

Kochan, Stephen. *Programming in C*. Englewood Cliffs, New Jersey: Hayden Book

第二章 初 启

2.1 引 言

本章介绍如何使用 **SETUP** 程序把编译程序装入磁盘，并设置编译程序的运行环境。

要启动和运行 C 编译程序，你需要做：

1. 后备磁盘（见 2.2 节）
2. 检查磁盘内容（见 2.3 节）
3. 从主配置盘上读出 **READMEDO.C** 文件，看一下本手册出版后，软件有何修改和扩充。
4. 运行 **SETUP** 程序，安装软件。
5. 阅读 2.8 节“编译程序命令行示范”或 2.9 节“实际操作阶段”，学习如何进行编译和链接。

一些 **MS-DOS** 过程会在本章中提到，特别要用到 **MS-DOS** 的 **SET** 和 **PATH** 命令给环境变量赋值，这些变量控制着编译程序环境，如果你不熟悉 **SET** 和 **PATH** 命令，或本章提到的其它 **MS-DOS** 过程，请参阅 **MS-DOS** 用户指南。

2.2 后备磁盘

打开系统盘以后，首先要用 **MS-DOS** 的 **COPY** 命令或 **D:SKCOPY** 实用程序生成工作副本。并保存原盘以备生成更多的工作副本。

2.3 磁盘内容

当你第一次打开你的编译程序包时，你应检查一下是否有了全部的软件。在主配置盘上可以看到一个文件名 **PACKING.LST**，这个文件存放在编译程序包的一个配置盘上。该文件列有组成编译程序软件的所有文件的清单及相应的描述，还有手册以及程序包中其它帮助你使用这些软件的一些资料。

通过 **PACKING.LST** 文件，你可以很快地对编译软件有个大概的了解，并且检查一下软件包是否完整。

注

Microsoft C 优化编译程序中包含的命名盘称作配置盘以区别于你用 **SETUP** 程序创建和标号的盘。

2.4 SETUP程序

SETUP 程序是一组 MS-DOS 批文件，负责自动安装编译程序。你在 1 号盘——主配置盘（该盘上可以还有其它文件）上找到 SETUP 程序。以下几小节介绍 SETUP 干什么以及如何启动 SETUP。

2.4.1 SETUP干什么

SETUP 程序执行下列任务：

- 把所有必要的文件复制到指定的目录或盘。
- 根据你的说明建立运行时库，每个库包括对数学，存储模型，和运行 SETUP 时所作的相容选择项的支持。在多数情形下，连接时只需一个库。
- 创建批文件 NEW-VARS.BAT，用来给环境变量置值，使得编译程序和链接程序能找到所需要的文件。
- 创建 NEW-COWF.SYS 文件，它记录对 CONFIG.SYS 文件中各 files（文件）和 buffers（缓冲）参数的合适的置值。

查看主配置盘上的 PACKING.LST 文件，可以看到 Microsoft C 优化编译程序的有关文件的完整清单。关于环境变量和 COWF.SYS 文件，详见 2.5.5 “编译程序环境”。

2.4.2 运行SETUP

按照下列两个步骤，使用 SETUP 程序：

1. 启动 SETUP 程序，并说明以下信息

- 程序所要使用的存储模型。
- 是否要和 Microsoft C 3.0 和 4.0 版本下建立的模块链接。
- 编译程序文件所要装入的各目录

2. 在 SETUP 提示下，键入程序要用到的浮点数学程序包的名字。

根据你对命令行所做的选择项和对提示的回答，SETUP 安装编译程序，并建立适合你所做的存储模型和浮点选择项的有关库。

注

运行 SETUP 程序时，你可能会得到“越出环境空间”的信息。此时，你应扩充环境空间。如果你使用的是 IBM PC-DOS 3.1 或者更早的版本，可用 SETENV 实用程序扩充环境。否则，你可用 SHELL 命令在 CONFIG.SYS 文件中说明一个更大的环境空间，后者重新启动系统。有关 SETENV，请看《Microsoft CodeView 和实用程序手册》，有关 SHELL 命令，请看 DOS 用户指南中关于 CONFIG.SYS 文件的叙述。

要熟悉运行 SETUP 要作的选择项，请看第 6 章“存储模型的使用”中有关存储模型及其使用的叙述，以及第 7 章“浮点数学运算的控制”中有关数学程序包及其使用的叙述。

2.4.2.1 选择 SETUP 命令行选择项

运行 SETUP 前, 把主配置盘插入软盘驱动器, 并使之成为当前驱动器。

要启动 SETUP, 打入如下形式的命令行, 并按 Enter 键:

SETUP base models [cmpat] [bindir] [includir] [libdir] [srcdir]

警告

SETUP 需要给定的参数包括若干目录名, 如果你所说明的目录不存在, 则 SETUP 自动建立。

注意不要给 SETUP 现存的目录名。除非你确知该目录下任何文件不与编译程序文件重名。因为 SETUP 装入重名文件时会冲掉现存文件。

要让 SETUP 对某个选择项使用适当的缺省, 只要省略该参数即可。某些任选的参数允许你键入问号 (?) 表示缺省。

下表解释 SETUP 命令行中你要给定的每一个参数:

参数	意义
base	用于安装的“基”目录的名字。该名必须以驱动器名(如c:)打头, 命令行中你所给定的其它所有目录都作为该基目录的子目录建立。
models	一个或多个字母, 用空格隔开, 告诉 SETUP, 程序将使用哪种(几种)存储模型, 键入 S 表示小模型, M 表示中模型, C 表示紧凑模型, L 表示大或超大模型。SETUP 根据你所键入的字母决定建立何种组合库。因为组合库很大, 你应该只说明程序所必要的存储模型。(如果你使用多种存储模型, 你可以使用非组合库, 而不必让组合库用尽磁盘空间, 详见 2.4.6)。
cmpat	如果想与 Microsoft c 4.0 版本兼容, 请键入 c40。如果要求与 4.0 版相容, SETUP 分别以 mSETARGV.OBJ 为名字装入 SETARGV.OBJ 文件的若干副本, 这里 m 是你所键入的表示 models 参数的一个字母(S, 小; M, 中; C, 紧凑; L, 大或超大)。如果你省去该参数, SETUP 只用通常的名字 SETARGV.OBJ 装入一个副本, 关于 SETARGV.OBJ 的叙述, 请见 2.5.4 “其它文件”。
bindir	base 的子目录, 其中要装入编译程序的可执行文件, 包括编译程序, 链接程序和实用程序。该参数必须以反斜杠 (\) 打头。 如果你省去该参数或者只打入一个问号 (?), SETUP 以 \BIN 子目录作为缺省值。
includir	base 的子目录, 其中要装入包含文件, 该参数必须以反斜杠 (\) 打头

如果你省去该参数或者只打入一个问号(?)，SETUP用\INCLUDE子目录作为缺省值。

libdir base的子目录，其中要装入库文件，该参数必须以反斜杠(\)打头。

如果你省去该参数或者只打入一个问号(?)，则SETUP用\LIB作为缺省值。

srcdir base的子目录，SETUP在其中拷入C启动文件(如需要)，你还要在其中拷入C源文件。如果指定，则该变元要以反斜杠(\)打头。

如果你打入一个问号(?)表示该参数，则SETUP用\SRC子目录作为缺省值。如果你不给出该参数，则SETUP不为源文件创建子目录。

除了你让SETUP创建的那些子目录外，SETUP还自动创建下列子目录：

- 子目录\TMP，编译程序在编译过程中用于存放暂时文件。
- 子目录\bindir\SAMPLE，SETUP在其中装入Microsoft C优化编译程序的有关演示程序。
- 一个或多个以base\srcdir\model为名字的子目录，其中model是一个表示存储模型的字母(S，小；M，中；C，紧凑；L，大或超大)。每个\srcdir\model子目录中含有一个名为VERSION.IMC的文件。只有当你创建了一个源文件子目录时，SETUP才建立这些子目录。

● 例子

SETUP S C:\ ???

本命令行告诉SETUP把编译软件装入根目录(\)的缺省子目录。缺省子目录是：\BIN用于编译程序和实用程序等可执行文件；\INCLUDE用于包含文件；\LIB用于库文件。没有为源文件创建子目录。且只建立小模型库文件。

SETUP C:\C S M C L C40 \BINDIR \INC \LIBS \SOURCES

本命令行告诉SETUP把编译软件装入C目录的给定子目录。可执行文件装入\C\BINDIR子目录；包含文件装入\C\INC子目录；库文件装入\C\LIBS子目录；源文件装入\C\SOURCES子目录；演示文件装入\C\BINDIR\SAMPLE子目录。建立了库文件，并针对存储模型装入各mSETARGV.OBJ文件。

2.4.2.2 选择浮点数学程序包

当你打入SETUP命令行后，SETUP显示一个消息告诉你它已经准备好建立组合库，然后按如下方式进行提示：

Do you wish to use the Emulator floating point math package? [y/n]

Do you wish to use the 8087/80287 floating point math package? [y/n]

Do you wish to use the Alternate floating point math package? [y/n]

在每个提示后：

- 如果你的程序要使用给定的浮点数学程序包并建立支持该程序包的组合库，请键入Y

或 y, 并按 Enter 键。因为组合库很大, 除非对那些你确知程序要用到的浮点程序包不要打入 .LY (如果你要使用多个浮点数学程序包, 为避免组合库占用大量磁盘空间, 你应用非组合库, 详见 2.4.6)

- 假如你不想使用给定的数学程序包, 打入 N 或 n 并按 Enter 键。

如果至少有一个提示你没有作出回答, SETUP 显示下列信息:

You did not specify a floating point math option and, thus, setup will not build any combined libraries.

Is this what you want? [y/n]

要回答这个提示:

- 打入 N 或 n 并按 Enter 键, 如果你还想建立运行时库。SETUP 回到先前关于浮点数学程序包的提示。

- 打入 Y 或 y 并按 Enter 键, 如果你不想让 SETUP 建立组合库。这种情况下, 你必须人工地安装非组合库。详见 2.4.6 “使用非组合库”。

2.4.2.3 在组合库中加入图形

SETUP 显示的最后一个提示询问你是否想在组合库中包括 Microsoft C 的图形功能:

Do you want the graphics package included in your combined libraries [y/n]?

- 打入 Y 或 y, 如果你将在程序中使用 Microsoft C 的图形功能, 并且希望把它们包括在 SETUP 建立的组合库中。

要回答这个提示:

- 打入 Y 或 y, 如果你将在程序中使用 Microsoft C 的图形功能, 并且希望把它们包括在 SETUP 建立的组合库中。

- 打入 N 或 n, 如果你不想在组合库中包括图形; 或者虽然你想在程序中使用图形功能, 却不想把它们包括在组合库中。请见 2.4.7 “使用 Microsoft C 图形库”。

2.4.2.4 建立库

如果你对至少一个数学程序包提示回答了 Y 或 y, SETUP 显示出它根据命令行中你选择的存储模型和 (在对提示符的响应中) 选择的浮点数学程序包建立的库的名字。SETUP 建立组合库为的是加快链接速度。当 SETUP 建好库以后, 它显示如下信息:

Library build complete.

Setup no longer needs the library sub-components and you do not normally need them to compile and link your C program. Do you want to delete them? [y/n]

如果你要用指定给 SETUP 以外的存储模型和浮点数学程序包, 你应保留非组合库。但是如果只使用组合库支持的存储模型和浮点数学程序包; 你就可以删除非组合库。如果你想删除非组合库, 输入 Y 或 y; 如果你想保留非组合库, 输入 N 或 n。

各个库建好后, SETUP 完成其余的安装工作, 包括建立 NEW-VARS.BAT 和 NEW-CO-NIF.SYS 文件。安装过程完成后, SETUP 显示如下信息:

Done!

2.4.2.5 安装后进行编译